1. SCOPE

MDI low rate (5 Kilo-bit/Sec) science data is produced by the image processor in a time frame asynchronous to other observing activities. Most low rate data products appear sometime between 1 and 2 minutes after the reference time associated with their set of frames. The data product suite also includes averaged data which is associated with a span of time, i.e. several reference times. This document defines the requirements for low rate science data product generation by the IP and data collection by the DEP. It outlines the method by which the data is transferred from the IP to the DEP, specifies the annotation information added by the DEP, and defines the telemetry format. This document is in general terms and does not define any of the processing steps in the IP required to produce data products.

2. OPERATIONAL OVERVIEW

The DEP executes repetitive observational operations under control of a module called the Sequence Control Routine (SCR). Sequences executed by SCR contain instructions defining the configuration of the instrument when images are taken with the camera. Sequences can also contain instructions that effect the operational flow, i.e., loop, test, and branch instructions. When SCR executes a take picture instruction, header information is sent to the camera electronics by the DEP. The header is sent to the Image Processor (IP) with the camera data. Specific words in this camera header can contain a pointer to the IP's Instruction Queue. If so, the IP executes the program at that address.

IP programs that realize the MDI Structure Program, produce a variety of data products by executing instructions that perform arithmetic operations on a series of pictures (images). Data products are eventually transferred to a buffer in the DEP/IP interface and the IP program execute a DEP interrupt instruction to inform the DEP that low rate science data products are available. The DEP acquires the data products from the buffer, formats them into packets, and transfers the data to the spacecraft telemetry interface.

The fundamental requirements (the prime directive) of the Solar Oscillations Investigation, the scientific program realized by MDI, demand highly repetitive time steps in the frame generation process. This requirement is realized in the DEP operational software and its interaction with the sequencer. Frame generation, IP processing, data transfer to the DEP, and telemetry output to the spacecraft operate across several time domains which are asynchronous. The most challenging aspect to the MDI software and firmware is managing the data flow while maintaining the cadence.

3. PACKET FORMAT

MDI science packets are 392 bytes long. The first several bytes are defined as shown in Table 1.

POSITION	LENGTH	DESCRIPTION
0	6	Local On-Board Time (LOBT)
6	1	Packet Format ID
7	1	Packet Number in Major Frame (0 - 23)
8	384	Data

Table 1 Science Data Packet Format

Packet Format ID is defined as follows:

* If the MSB is 0, the data is normal IP generated Data Products and the low seven bits are the segment number of the current data product.

* If the MSB is 1, the data is a special format. The following special formats have been defined:

  FF: Fill
  FE: Limb Tracker Sample
  FD: ROM Dump
  FC: Motor Currents

4. DATA STREAM

The SOHO data system is packetized. 24 times per major frame (approximately every 15 seconds), the spacecraft requests an MDI low rate science packet. There is no correlation between packet size and data product size. To optimize the utilization of the data stream, data product can be nested within and/or span packets. Table 2 shows the general format for low rate science data products.

NAME	LENGTH (BYTES)	DESCRIPTION
SYNC	2	To identify Data Products in stream (AA55)
DPID	4	Data Product Identifier
REFT	6	Reference Time (from CCD Header)
LENG	2	Data Product length (Words)
INFO	8	Additional Data Product Information
DATA	Variable	Data

Table 2 Low Science Data Product Format

5. BASELINE DATA PRODUCTS

The current baseline includes data products generated once per minute[1] and others which are time averages over several minutes. The following list includes all low rate data products that will be generated by the initial release of the flight firmware:

* Every Minute:
  SEQD → Sequence Definition Data (Data Product 0)
  LOIV → 64 element velocity
  LOIC → 64 element continuum
  GOLF →32x32 Magnetic Proxy
  WAxx → Some Number of compressed weighted averages

* Time Averages:
  LIMB → 15000 Elements 10 min.
  LRCI → Low Resolution Continuum Image 16K 20 min.

6. OBSERVING CYCLE FORMAT

Each sequence generates the same Low Rate Science Data Products[2]. During each minute, the data product set includes the same products, i.e. the data stream contains all 1minute cadence data products plus 1/10 of LIMB and 1/20 LRCI. Both LIMB and LRCI are compressed by the IP firmware. The firmware compresses these data products in sections whose size is given by their output telemetry allocation. For example, LIMB is allocated 1000 words per minute[3]. Each minute the IP compresses LIMB until the compressed output fills 1000 words; thus the number of uncompressed data elements in each minute sample varies. If a time averaged sample does not compress well, data will be lost at the end.

[1] One minute is the nominal observing cycle. This may be adjusted later in the mission. In this document, all references to sequence timing use minutes.
[2] This is true by convention only. There is nothing in the DEP software or IP firmware to force a sequence to produce only 1 set of data products; however, it is inconsistent with the prime directive not to do so.
[3] These numbers are arbitrary. Real values are TBD.

7. DEP/IP INTERACTIONS

7.1. OVERVIEW

Underlying the design of the low rate science data function are these two facts:

* The IP must have detailed knowledge about the construction of data products and where to place them in the DEP/IP data buffer.

* The DEP needs to know when data products are available for down link and how to format data products into the telemetry stream.

The respective roles of the two processors are extended from these two points until the entire required functionality is encompassed.

7.2. DATA TRANSFER FROM THE IP TO THE DEP

All data transfer from the IP to the DEP using the 5KB buffer are always the same regardless of data content. The buffer passed to the DEP will contain a directory section and a data section. The directory section is 256 words long, containing sufficient space for a directory header and 31 directory entries. The format is shown in Tables 3-1 through 3-3.

WORD	DESCRIPTION
0 . 7	DIRECTORY HEADER
8 . 255	DIRECTORY ENTRIES . .
256 . . . 32767	DATA PRODUCTS . . .

Table 3-1 IP/DEP DATA TRANSFER FORMAT

WORD	DESCRIPTION
0	NUMBER OF DATA PRODUCTS
1	REFERENCE TIME
2	.
3	.
4	RESERVED WORDS
5	.
6	.
7	.

Table 3-2 DATA PRODUCT DIRECTORY HEADER

WORD	DESCRIPTION
0	DATA PRODUCT IDENTIFIER (DPID)
1	.
2	ADDRESS
3	LENGTH
4	SEQUENCE COUNT
5	RESERVED WORDS
6	.
7	.

Table 3-3 DATA PRODUCT DIRECTORY ENTRY

7.3. IP IMPLEMENTATION

Elements such as reference time and sequence count will not have special meaning in the IP. The above will be accomplished within the existing concepts of the IP firmware. The responsibility to see that the directory header and directory entries are correct lies to a large extent with the ground segment.

7.3.1. DATA PRODUCT DIRECTORY HEADER

There will be a new IP instruction, **ILRD**, which initializes low rate data product generation. This instruction includes a user specified IP register. This register is the start of a dedicated 8 word register set in which the directory header is constructed. The first register in the set is set to zero and incremented each time a move low rate data product instruction is executed. The next 3 registers contain the reference time, while the last 4 are available for whatever. It is the responsibility of the IP program initiated by a camera frame to get the reference time into the appropriate register. To further facilitate this and other data annotation functions, there will be another new IP instruction, **MDPK**, which extracts the low bytes of a series of words and packs the results into a register set.

7.3.2. DATA PRODUCT DIRECTORY

An IP program moves data products into DEP/IP buffer by executing **WDPB** or **CDPB** instructions. To implement the production of the data product directory, these instructions will be modified to include the DPID and a register number. The additional data product information (INFO in Table 2), is a set of four registers that begin with the register specified in the instruction. When the IP transfers the data product to the buffer, it will increment the data product count and create a data product directory entry. (How and where directory entries are created is an implementation detail). The role of the four registers is to allow the program the flexibility to annotate data products. The only define use for this information is the sequence count in time averaged data.

7.3.3. DATA PRODUCT FINALIZATION

Upon transfer of the last data product in a group, the IP program must execute a new IP instruction, **FLDB**, to finalize the data product buffer and a **SDPI** to set a DEP interrupt. **FLDB** transfers the directory header to the buffer and performs any other finalization tasks that may be required by the IP implementation.

8. UNDERLYING PRINCIPLES

The design implementation is based on a few principles. This section elaborates on the terse staement of these principles given in 7.1 so that the intention behind the design is not lost.

8.1. WCS STABILITY

It is relatively difficult and risky to change the writable control store load after launch. The design therefore locks into the WCS only those processes that are least likely change. For example,the velocity and intensity algorithms are built-in to the firmware with NO parameterization supplied by the instruction other than the input and output buffer addresses and the lookup table address. The generation of data products is handled by individual instructions that create data products in memory. Move data product instruction then transfer the data to the buffer. The transfer to the buffer, location in the buffer, size of the data product, and the ID are all parameterized rather than built-in as there is every reason to expect variation in these parameters over the course of the mission.

8.2. REQUIRED KNOWLEDGE

Since the transfer of data products to the buffer is not built into the firmware, per se, it must be expressed in the IP program executing in the instruction queue. Since these instructions contain the information about the location and length of the data products, it is a small step to pass this information to the DEP as a sort of directory. This approach alllows the DEP to format data products generated by the IP with no prior or independent knowledge. In other words, the DEP can create a low rate science stream based simply on the demand presented to it by the IP and eliminates the need for duplicate specification of IP data products in the DEP.

8.3. INHERENT POWER OF THE GROUND SEGMENT

The approach outlined above requires that several inherently coupled processes be explicitly coupled in IP programs. For example, the program must contain instructions to move the reference time to a register set that will be used to create the directory header, and the instruction that initializes the header must also specify this register set. While at first this may seem redundant, the notion is that the experiment writer has immense compute power available to him on the ground. The integerity of IP programs can either be built-in the compiler and/or linker or can be check by some "lint" like utility. Building too much detail about data product transfer to the DEP into the WCS load could stifle flexibilty later in the mission.

9. TYPICAL LOW RATE SCIENCE PROGRAM

The following summarizes the flow of a typical low rate science data program:

Camera frame starts the program:

```
Move reference time to register set
Move info from CCD header to made SEQD
Extract CCD Temperatures
Create various 1 minute data products
                ...
                ...
Is it a new 10 minute interval?
        Yes.... Scale and transfer the accumulated LIMB sum to staging area
                Clear accumulation buffer and reset count
Is it a new 20 minute interval?
        Yes.... Scale and transfer the accumulated LRCI sum to staging area
                Clear accumulation buffer and reset count
Initialize low rate buffer directory
Transfer 1 minute data products
                ...
                ...
Extract, compress, and transfer next piece of LIMB
Extract, compress and transfer  next piece of LRCI
Finalize the buffer transfer
Interrupt the DEP
```

Return